

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-39. (Cancelled)

40. (New) A computer implemented method for replicating a software application in a multi-computer architecture cluster, the computer implemented method comprising:

executing the software application on a first computer of the cluster forming a primary node;

replicating the resources associated with the software application on at least one other computer of the cluster forming a secondary node, the resources associated with the software application on the secondary node comprising a virtual memory and calling stack of each process affected, system resources comprising inter-process communication, network connection, and data written on disks; and

updating the replicated resources incrementally, using a dynamic introspection mechanism supplying a structure of the software application to be replicated and also supplying a dynamic graph of the resources and dependencies implemented.

41. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

creating and maintaining a dependency tree, based on the dynamic graph, supplying, at all times, information on the replicated resources.

42. (New) The computer implemented method according to claim 40, wherein replicating the resources associated with the software application further comprises:

checkpointing the resources on at least one secondary node, wherein the checkpointing having an adjustable period.

43. (New) The computer implemented method according to claim 42, wherein replicating the resources associated with the software application further comprises:

capturing the resources on the primary node to create captured resources;
transferring the captured resources over the network to at least one secondary node; and
restoring the captured resources on the at least one secondary node.

44. (New) The computer implemented method according to claim 42, wherein replicating the resources further comprises:

optimizing the checkpointing.

45. (New) The computer implemented method according to claim 44, wherein the checkpointing is incremental.

46. (New) The computer implemented method according to claim 44, wherein the checkpointing is discriminating.

47. (New) The computer implemented method according to claim 42, wherein the checkpointing further comprises at least one of the following:

processing a synchronization barrier;
managing resources;
managing system resources; and
managing process resources.

48. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

replicating applicative data files between the primary node, whereon the software application is run, and a stand-by node.

49. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the

computers of the cluster, called the primary node, while other computers of the cluster are called secondary nodes, wherein ensuring functional continuity further comprises:

- replicating the software application on at least one of the secondary nodes to provide at least one clone of the application,
- updating the at least one clone, and
- responsive to detecting an event affecting the primary node, switching from the software application being executed on the primary node, to the software application being executed on the at least one clone.

50. (New) The computer implemented method according to claim 49, wherein replicating the software application is of a holistic nature.

51. (New) The computer implemented method according to claim 49, wherein updating the at least one clone further comprises updating the clones of the application.

52. (New) The computer implemented method according to claim 49, wherein ensuring functional continuity further comprises supervising a state of the resources necessary to operate the software application.

53. (New) The computer implemented method according to claim 49, wherein detecting an event affecting the primary node further comprises:

- responsive to detecting an event affecting the primary node, electing a clone to be substituted for the primary node of the software application, wherein the secondary node on which the clone elect is installed becomes a new primary node.

54. (New) The computer implemented method according to claim 53, wherein replicating the resources further comprises:

- recording, on the at least one clone, messages received by the primary node, the messages being injected into the clone elected as the new primary node when switching.

55. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

optimization of information processing resources by load sharing and dynamic process distribution.

56. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

performing non-interruptive maintenance by process relocation upon request, over a data-processing resource network.

57. (New) The computer implemented method according to claim 40, wherein replicating the resources further comprises:

preserving applicative context in a mobile application.

58. (New) A multi-computer system for ensuring functional continuity, capable of running, on at least one computer, at least one software application, the multi-computer system comprising:

a memory comprising a set of instructions;

a processor connected to the memory, capable of executing the set of instructions to implement a method comprising:

ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the computers of the cluster, called a primary node, while other computers of the cluster are called secondary nodes, wherein ensuring functional continuity further comprises:

replicating the software application on at least one of the secondary nodes to provide at least one clone of the application, wherein replicating the software application is of a holistic nature;

updating the at least one clone, and

responsive to detecting an event affecting the primary node, switching from the software application being executed on the primary node, to the software application being executed on the at least one clone.